

Title: Using machine learning methods to predict experimental high-throughput screening data

Chérif Mballo and Vladimir Makarenkov^{*}

*Département d'informatique, Université du Québec à Montréal,
C.P. 8888, Succursale Centre-Ville, Montreal (QC) H3C 3P8 Canada*

Emails: mballo.cherif@courrier.uqam.ca and makarenkov.vladimir@uqam.ca

^{*} Corresponding author. Tel: (+1) 514-987-3000, Ext: 3870; Fax: (+1) 514-987-8477;
E-mail: makarenkov.vladimir@uqam.ca

Abstract. High-throughput screening (HTS) remains a very costly process notwithstanding many recent technological advances in the field of biotechnology. In this study we consider the application of machine learning methods for predicting experimental HTS measurements. Such a virtual HTS analysis can be based on the results of real HTS campaigns carried out with similar compounds libraries and similar drug targets. In this way, we analyzed *Test assay* from McMaster University *Data Mining and Docking Competition* [1] using binary decision trees, neural networks, support vector machines (SVM), linear discriminant analysis, k -nearest neighbors and partial least squares. First, we studied separately the sets of molecular and atomic descriptors in order to establish which of them provides a better prediction. Then, the comparison of the six considered machine learning methods was made in terms of false positives and false negatives, method's sensitivity and enrichment factor. Finally, a variable selection procedure allowing one to improve the method's sensitivity was implemented and applied in the framework of polynomial SVM.

Key words: CART, decision trees, drug target, hit, k -nearest neighbors (kNN), linear discriminant analysis (LDA), neural networks (NN), partial least squares (PLS), ROC curve, sampling, support vector machines (SVM), virtual high-throughput screening.

1 Introduction

High-throughput screening (HTS) is an efficient but still very expensive technology intended to automate and accelerate the discovery of pharmacologically active compounds (i.e., potential drug candidates). A typical HTS campaign involves testing a large number of chemical compounds in order to generate *in vivo* primary hits, which may be promoted to secondary hits, and then to leads, after additional experiments. Several data correction procedures have been recently proposed to address the needs of experimental HTS campaigns [2-5]. But, the field of HTS is also very suitable for applying machine learning methods, and the development of an accurate procedure allowing one to predict *in silico* the compound's activity would be of great benefit for the pharmaceutical industry.

A number of recent studies have discussed the application of machine learning methods in HTS. Thus, Briem and Günther [6] carried out the support vector machines (SVM), artificial neural networks (NN) and k -nearest neighbors (kNN) methods with a genetic algorithm-based variable selection and recursive partitioning in order to distinguish between kinase inhibitors and other molecules with no reported activity on any protein kinase. Using the majority vote for all tested techniques, the latter authors concluded that NN provided the best prediction of experimental results, followed by SVM. On the other hand, Müller et al. [7] described an application of SVM to the problem of assessing the "drug-likeness" of a compound based on a given set of molecular descriptors. The authors concluded that in the drug-likeness analysis a polynomial SVM with a high polynomial degree ($d = 11$) allows for a very complex decision surface which could be used for prediction. Burton et al. [8] applied recursive partitioning based on decision trees, for predicting the CYP1A2 and CYP2D6 inhibition. The latter authors noticed that with a set of mixed 2D and 3D descriptors, the trees gained

2 to 5% on each accuracy parameter compared to the 3D descriptors used alone. Harper and Pickett [9] reviewed recent literature dealing with the application of data mining techniques in the field of HTS. Plewczynski et al. [10] showed that an SVM model can sometimes achieve classification rates up to 100% in evaluating the activity of compounds with respect to specific targets. The latter authors concluded that the obtained sensitivities for all considered protein targets exceeded 80%, and the classification performance reached 100% for the selected targets. Simmons et al. [11] conducted a study in which 10 machine learning methods were used to develop classifiers on a data set derived from an experimental HTS campaign, and compared the method predictive performances in terms of false negative and false positive error profiles. The set of descriptors considered by Simmons et al. [11] consisted of 825 numerical values, representing 55 possible atom-type pairs mapped to 15 distance ranges. Alternatively, Simmons et al. [12] described an ensemble-based decision tree model to virtually screen and prioritize compounds for acquisition. Butkiewicz et al. [13] compared NN, SVM and decision trees in a specific QSAR approach. The latter authors applied the three above-mentioned machine learning methods for screening *in silico* potentiators of metabotropic glutamate receptor subtype 5 (mGluR5). Butkiewicz et al. [13] concluded that SVM performed slightly better than NN, which had in turn an advantage over decision trees. Fang et al. [14] presented an effective application of SVM in mining HTS data in a type I methionine aminopeptidases (MetAPs) inhibition study. SVM was applied on a compound library of 43,736 small organic molecules. Fang et al. [14] discovered that 50% of the active molecules could be recovered by screening just 7% of compounds of the test set. Each of these studies was conducted in particular statistical (e.g., sampling strategies) and HTS (e.g., kind of HTS data, proportions of hits in the data and available descriptors) contexts, making the comparison of the obtained results a very difficult task.

In this article we examine an experimental HTS assay from the McMaster University *Data Mining and Docking Competition* [1] by means of the following machine learning methods: binary decision trees (CART), Neural Networks (NN), Support Vector Machines (SVM), Linear Discriminant Analysis (LDA), Partial Least Squares (PLS) and *k*-nearest neighbors (kNN). After the description of the considered McMaster assay, we will test separately the sets of molecular and atomic descriptors characterizing the selected compounds in order to determine which of them provides a better prediction. The proportion of false positive and false negative hits in the test sets and the sensitivity performances of each considered machine learning method will be assessed and discussed in detail. The competing machine learning methods will be also compared in terms of identification of hits (i.e., the set of 96 average hits disclosed by the competition organizers; for more details see Elowe et al. [1]) and enrichment factor. Finally, we will describe a stepwise procedure for adding and removing explanatory variables (i.e., descriptors) from the set of considered descriptors in order to improve the prediction performance.

2 Data description

In this study, *Test assay* of the McMaster Data Mining and Docking Competition [1] (<http://hts.mcmaster.ca/Downloads>) was considered. This *Test assay* consisted of a screen

of compounds intended to inhibit the *Escherichia coli dihydrofolate reductase* (DHFR). Each compound of the assay was screened twice: two copies of 625 plates were run through the screening machine. The screen performed at the McMaster HTS Laboratory was carried out in duplicate using the Beckman-Coulter Integrated Robotic System (using 96-well polystyrene, clear, flat bottom, non-treated, non-sterile plates). The compounds were located on rectangular plates with 12 columns and 8 rows (Fig. 1). Three different types of controls: high, low and reference controls, were used in the screen (Fig. 1) in order to normalize the raw HTS measurements on a plate-by-plate basis [1]. The values of the first and last column (the columns 1 and 12) containing the control compounds were not considered any longer after the normalization of the raw data was performed.

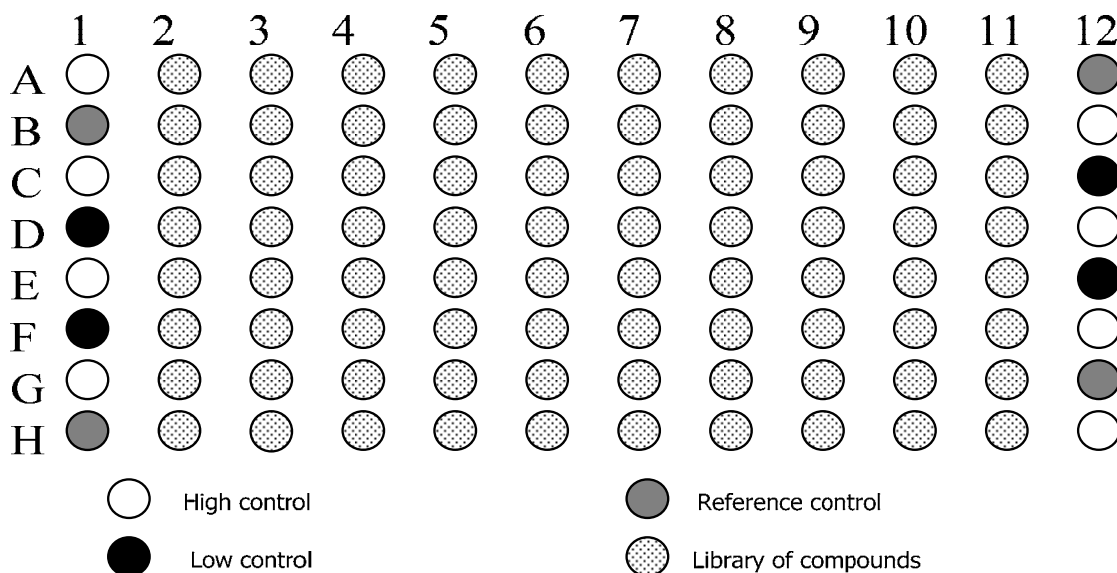


Fig. (1). Plate layout of the McMaster *Test assay* (according to the technical documentation of the McMaster University HTS laboratory).

A compound was declared as an *average hit* if it reduced the average percent residual activity of DHFR below the cut-off value of 75% of the average residual enzymatic activity of the high controls. Of the 50,000 screened compounds, a total number of 96 *average hits* were identified [1]. The first set of screened data with the corresponding experimental results was publicly released and called *Training data set*. Then, computational chemists and data analysts were challenged to predict the compounds activity in the second data set called *Test set*.

Molecular descriptors are numerical values that describe the structure and shape of molecules (i.e., compounds), helping predict their activity and properties in complex experiments. Each molecular descriptor takes into account one part of the whole chemical information contained into the real molecule [15]. Atomic descriptors are 3D motifs produced from atoms belonging to relevant cavity surfaces [16]. As examples of atomic and molecular descriptors included in the combined set of descriptors examined in this study, mention: molecular weight, number of H-accepting and H-donating atoms, number of rotatable bonds, topologic polar surface area and two flavors of *log* of the octanol/water partition coefficient (*ClogP* and *SlogP*). In this study, we originally

considered 209 molecular descriptors and 825 atomic descriptors whose values were calculated for all the compounds of *McMaster Test data set*. The molecular descriptors were calculated using the MOE package [17] developed by *Chemical Computing Group* and the atomic descriptors were obtained using the software written by Dr. Kirk Simmons (see Simmons et al. [11]). Then, we selected the 209 atomic descriptors, among the 825 originally considered, whose values provided the highest correlation with the normalized average biological activity of the library compounds in order to compare them to the 209 molecular descriptors calculated by MOE [17]. Afterwards, we carried out the stepwise variable selection procedure [18] based on the linear regression, and including both forward and backward variable selection, using the MATLAB package (Version R2008a) in order to select the best atomic and molecular descriptors to be included in the combined data set of explanatory variables.

In *McMaster Test data set*, we have only 96 active compounds (i.e., 96 primary hits). We increased the original cut-off value of 75% to 81.811% in order to obtain exactly 1% of hits (i.e., 500 hits) in our data (i.e., we considered as hits all the compounds that reduced the average residual enzymatic activity of DHFR below the new cut-off value of 81.811%). If only the 96 average hits were considered, the proportion of the active compounds in the whole data set would be very marginal (0.192% of the total number of compounds), leading to a very unbalanced data set. It is worth noting that unbalanced data sets are one of the biggest hurdles to overcome in the field of HTS. An alternative to this approach would be shrinking the set of inactive compounds to insure a better hit/no hit balance but we did not consider this possibility. Thus, 500 active compounds, including 96 average hits and 404 *additional active compounds*, and 49,500 inactive compounds from *McMaster Test data set* were examined in our simulations.

3 Sampling strategy

In modeling unbalanced data sets, typical in HTS, the classifiers (e.g., machine learning methods) tend to predict that all the experiments have the outcome of the majority class (i.e., all the compounds are inactive) and miss entirely the minority class (i.e., active compounds). An appropriate sampling strategy is very important for such kind of data. In this way, it is desirable to reduce the amount of data presented to the data mining techniques.

As the inactive compounds are dominant in our data (49,500 inactive compounds versus 500 active compounds), we always consider all the active compounds and randomly sample the inactive ones. We considered the three following ratios for the training and test sets used in machine learning (each sample was divided into two independent parts: the training set to build the model and the test set to evaluate the model's performance):

- **Ratio 1:** 85% for the training set and 15% for the test set (these percentages were applied to the inactive compounds, average hits and additional active compounds). For the whole dataset, we obtained the following training set and test set proportions:

Training set: 42,075 inactive compounds (i.e., 85% of 49,500 inactive compounds), 425 active compounds, including 81 average hits (i.e., 85% of 96 average hits), and 344 additional active compounds (i.e., 85% of 404 additional active compounds).

Test set: 7,425 inactive compounds (i.e., 15% of 49,500 inactive compounds), 75 active compounds, including 15 average hits (i.e., ~15% of 96 average hits), and 60 additional active compounds (i.e., ~15% of 404 additional active compounds).

- **Ratio 2:** 70% for the training set and 30% for the test set.

Training set: 34,650 inactive compounds, 350 active compounds, including 66 average hits, and 284 additional active compounds.

Test set: 14,850 inactive compounds, 150 active compounds, including 30 average hits and 120 additional active compounds.

- **Ratio 3:** 50% for the training set and 50% for the test set.

Training and test sets: 24,750 inactive compounds, 250 active compounds, including 48 average hits, and 202 additional active compounds.

Each sample of the training (or test) set collection included all the active compounds and the randomly chosen inactive compounds. The inactive compounds were selected with replacement from the whole set of inactive compounds. For each selected ratio and each machine learning method being tested, we carried out 100 repeated calculations (each calculation provided a different model) and computed the average results for 100 models for the following statistics: false positive and false negative rates, sum of errors, and sensitivity and specificity of the method.

Alternatively, the ensemble sampling approach considered in Simmons et al. [11] could be used to create the enriched samples, but here we did not proceed in this way.

Table 1. Training set (85%) versus test set (15%).

| <i>Hit/No hit ratio</i> | <i>Sample size</i> | <i>Training size</i> | <i>Test size</i> | <i>% of hits</i> |
|-------------------------|--------------------|----------------------|------------------|------------------|
| (1:1) | (500:500) | (425:425) | (75:75) | 50.00 |
| (1:2) | (500:1000) | (425:850) | (75:150) | 33.33 |
| (1:3) | (500:1500) | (425:1275) | (75:225) | 25.00 |
| (1:4) | (500:2000) | (425:1700) | (75:300) | 20.00 |
| (1:5) | (500:2500) | (425:2125) | (75:375) | 16.67 |

Table 2. Training set (70%) versus test set (30%).

| <i>Hit/No hit ratio</i> | <i>Sample size</i> | <i>Training size</i> | <i>Test size</i> | <i>% of hits</i> |
|-------------------------|--------------------|----------------------|------------------|------------------|
| (1:1) | (500:500) | (350:350) | (150:150) | 50.00 |
| (1:2) | (500:1000) | (350:700) | (150:300) | 33.33 |
| (1:3) | (500:1500) | (350:1050) | (150:450) | 25.00 |
| (1:4) | (500:2000) | (350:1400) | (150:600) | 20.00 |
| (1:5) | (500:2500) | (350:1750) | (150:750) | 16.67 |

Table 3. Training set (50%) versus test set (50%).

| <i>Hit/No hit ratio</i> | <i>Sample size</i> | <i>Training and test size (same size)</i> | <i>% of hits</i> |
|-------------------------|--------------------|---|------------------|
| (1:1) | (500:500) | (250:250) | 50.00 |
| (1:2) | (500:1000) | (250:500) | 33.33 |
| (1:3) | (500:1500) | (250:750) | 25.00 |
| (1:4) | (500:2000) | (250:1000) | 20.00 |
| (1:5) | (500:2500) | (250:1250) | 16.67 |

Tables 1, 2 and 3 report the selected sample sizes and the considered proportions of hit/no hit compounds. The notation ($n:m$) indicates that, in the sample, we have n active and m inactive compounds. The presented ratios of the hits and no hits were adopted following the sampling strategy used in Simmons et al. [11].

4 Machine learning methods

The main goal of HTS is an accurate prediction of active compounds. Thus, HTS is a very natural field for applying machine learning methods. In this section, we outline the decision trees, neural networks, support vector machines, k -nearest neighbors, linear discriminant analysis and partial least squares methods being tested in this study. In our experiments, the R2008a version of the MATLAB package, including all these methods, was used to generate the results.

Classification and Regression Trees (CART)

Decision trees [19] are very popular in machine learning. A decision tree is a tree-like structure with a set of attributes to be tested in order to predict the output. In this study, we used the CART [19] method with the “*Classregtree*” function and the Gini splitting criterion.

Artificial Neural Networks (NN)

Artificial neural networks [20] (NN) have been widely employed in data mining as a supervised classification technique. The performance of an NN depends on both, the selected parameters and the quality of the input data. Larger numbers of *neurons* in the hidden layer give the network more flexibility. To improve the accuracy, one can also increase the number of *epochs* (i.e., number of complete passes by training data set). In this study, we used the backpropagation algorithm introduced by Rumelhart et al. [21]. It carries out learning on a multi-layer feed-forward neural network through an iterative process with a set of training samples. For each training sample, the weights were adjusted to minimize the mean squared error between the desired and obtained outputs. We used the MATLAB function “*Traindx*” (with adaptive learning rate). The NN performances were assessed using the mean squared error. The number of *epochs* was set to 5,000 and the number of neurones in the hidden layer varied from 50 to 250 (with the step of 50), depending of the sample size.

Support Vector Machines (SVM)

Support Vector Machines (SVM) were introduced by Vapnik [22]. They have been extensively applied in different fields including pattern recognition. SVM classification proceeds by computing a separating hyperplane between two groups of data while maximizing the distance from this hyperplane to the closest data points. This is equivalent to solving a quadratic optimization problem. The “*Svmtrain*” function of MATLAB was used in this study with linear, polynomial and rbf (radial basis function) kernel functions to find the optimal separating hyperplane. The degree of the polynomial kernel was set to 4. We used the quadratic programming, as method for the linear and

polynomial kernels. For the Gaussian Radial Basis Function kernel, the σ scalar was set to 1. All the other SVM parameters were the default MATLAB parameters.

***k*-Nearest Neighbors (kNN)**

K-nearest neighbors [23] is a supervised machine learning method where a new object is classified based on closest training examples in the feature space. The classification uses the majority vote criterion. This method carries out the neighbourhood classification as the prediction value for the new instance. It computes the minimum distance from the object to the training samples to determine the *k* nearest neighbors (the *k* closest points represent the voters). In the MATLAB package, this method is implemented in the “*Knnclassify*” function. Two distances, the “*Euclidean*” distance and the “*Cityblock*” distance (i.e., the sum of absolute differences), were tested. In our experiments, we varied the value of *k* from 1 to 4 and found that the best results were generally obtained with *k* = 1 (i.e., when we increase *k*, the sum of errors also increases) and with the “*Cityblock*” distance.

Linear Discriminant Analysis (LDA)

Linear discriminant analysis [24] is used to determine which variables discriminate better between two or more groups. LDA is closely related to the analysis of variance and the regression analysis. This method is implemented in MATLAB R2008a in the function “*Classify*”.

Partial Least Squares (PLS) regression

Partial least squares regression [25, 26] is a statistical method used to find fundamental relationships between two matrices including the response and explanatory variables. The main goal of PLS is to find the hyperplanes of maximum variance separating these variables. The PLS methods are known as bilinear factor methods because both the response and explanatory variables are projected into a new space. A PLS method will try to find the best multidimensional direction in the space of explanatory variables that accounts for the maximum multidimensional variance direction in the response space. In this study, the MATLAB function “*Plsregress*” was used. To carry out this algorithm, we first normalized the data using the MATLAB function “*Zscore*”. For a given matrix **M**, *Zscore*(**M**) returns a centered and scaled version of **M**.

5 Comparison of molecular and atomic descriptors

In HTS, a large number of chemical descriptors is usually available. The goal of the variable selection is to identify the subset of measured variables that best characterize the system under study. For the considered McMaster *Test data set*, we computed the values of 209 molecular descriptors (i.e., variables) using the MOE package, while the values of 825 atomic descriptors were calculated using a program written by Dr. Simmons [11]. The molecular structures were used to compute the atom-pair descriptors and all possible “*atom type - distance - atom type*” combinations were considered (for more details, see Simmons et al. [11]). Subsequently, we calculated the correlation coefficient between each of the obtained 825 atomic descriptors and the quantitative response variable (i.e.,

normalized average compound's activity), and 209 atomic descriptors associated with the highest values of the correlation coefficient were retained for further analysis. Thus, both molecular and atomic data sets being tested contained the same number of descriptors. We then transformed the quantitative response variable into a binary variable as follows: all the values lower than or equal to 81.811 were set to 1 (i.e., they correspond to the active compounds or hits), whereas all the remaining values were set to 0 (i.e., they correspond to the inactive compounds). Then, we tested four machine learning methods: Neural Networks, SVM with linear and polynomial kernels and Linear Discriminant Analysis. These methods were carried out separately on the sets of molecular and atomic descriptors in order to identify which of them provide a better prediction of the hit/ no hit outcomes.

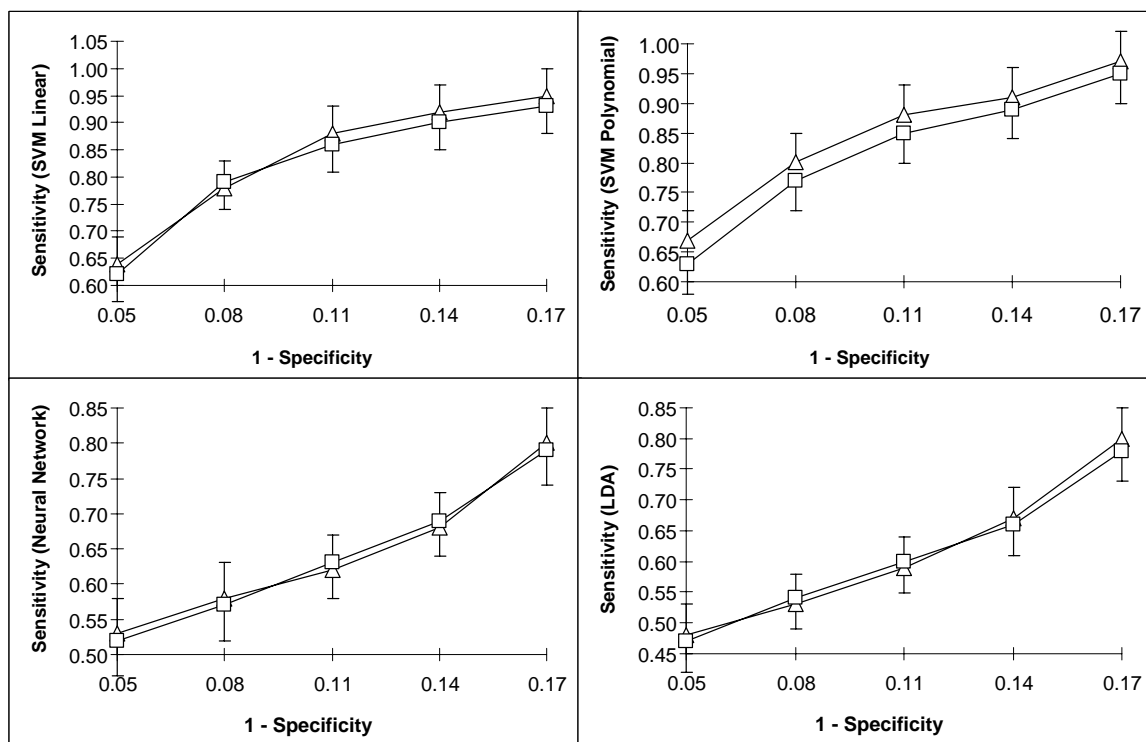


Fig. (2). Molecular versus atomic descriptors comparison for linear and polynomial Support Vector Machines, Neural Networks and Linear Discriminant Analysis. Molecular descriptors are depicted by triangles and atomic descriptors by squares. The proportions of hits and no hits and the sizes of training and test samples were those reported in Table 1. The five points of the x-axis correspond, from left to right, to the following hit/no hit ratios: (1:5), (1:4), (1:3), (1:2) and (1:1). Positive error bars are shown for molecular descriptors and negative error bars for atomic descriptors. The lengths of negative error bars for molecular descriptors and positive error bars for atomic descriptors were very similar to the corresponding opposite direction bars shown in the figure.

We considered the proportions of hits and no hits and the sizes of training and test samples reported in Table 1. For each method, the average results computed over 100 repeated calculations were reported (Fig. 2). At each such a calculation, a different set of inactive compounds was selected, whereas the active compounds did not change throughout the process. In each experiment, we assessed the average rates of false positives and false negatives and calculated the specificity (Sp) and the sensitivity (Se) of the methods according to Equation 1:

$$Sp = \frac{TN}{TN + FP} \quad \text{and} \quad Se = \frac{TP}{TP + FN}, \quad (1)$$

where TP is the number of true positives, FN – number of false negatives, TN – number of true negatives and FP – number of false positives. The sensitivity and specificity are widely used to assess the performances of machine learning methods. The 95%-confidence intervals for the sensitivity were also computed to assess the reliability of this estimate. They are denoted as $CI_{Se(95\%)}$ and defined as follows (Equation 2):

$$CI_{Se(95\%)} : Se \pm 1.96 \sqrt{\frac{Se(1 - Se)}{TP + FN}}. \quad (2)$$

Figure 2 illustrates the performances of the four competing machine learning methods using ROC curves [27]. Tables 4 and 5 report the 95%-confidence intervals for sensitivity for the molecular and atomic descriptors, respectively. While examining the curves for the NN and LDA methods (Fig. 2), one can notice that the atomic and molecular descriptors yielded very similar results in terms of predicting active compounds. But, with linear SVM and, in particular, with polynomial SVM which provided the best overall performance, the molecular descriptors usually outperformed the atomic ones. For instance, for the polynomial SVM method and for both molecular and atomic descriptors the true positive rate varied between 67 and 97%, while maintaining the false positive rate in the 5-17% range. These trends were also noticeable for the confidence intervals presented in Tables 4 and 5. For a given hit/no hit ratio, the obtained confidence intervals for the linear and polynomial SVMs were very similar. This was also the case of the confidence intervals found for the NN and LDA methods. For a given method and hit/no hit ratio, the confidence intervals were very similar for both molecular and atomic descriptors (see the columns of Tables 4 and 5). This indicates that the differences between the predictions provided by the molecular and atomic descriptors were not significant.

Table 4. 95%-confidence intervals for sensitivity (computed according to Equation 2) obtained with the linear SVM, polynomial SVM, NN and LDA methods for the molecular descriptors.

| <i>Hit/No hit ratio</i> | <i>Linear SVM</i> | <i>Polynomial SVM</i> | <i>Neural Networks</i> | <i>LDA</i> |
|-------------------------|-------------------|-----------------------|------------------------|-------------|
| (1:5) | [0.53,0.75] | [0.55,0.76] | [0.42,0.64] | [0.37,0.59] |
| (1:4) | [0.69,0.87] | [0.71,0.88] | [0.47,0.69] | [0.42,0.64] |
| (1:3) | [0.81,0.95] | [0.80,0.95] | [0.51,0.73] | [0.48,0.70] |
| (1:2) | [0.86,0.98] | [0.84,0.97] | [0.57,0.79] | [0.56,0.78] |
| (1:1) | [0.90,1.00] | [0.92,1.00] | [0.71,0.89] | [0.71,0.89] |

Table 5. 95%-confidence intervals for sensitivity (computed according to Equation 2) obtained with the linear SVM, polynomial SVM, NN and LDA methods for the atomic descriptors.

| <i>Hit/No hit ratio</i> | <i>Linear SVM</i> | <i>Polynomial SVM</i> | <i>Neural Networks</i> | <i>LDA</i> |
|-------------------------|-------------------|-----------------------|------------------------|-------------|
| (1:5) | [0.51,0.73] | [0.52,0.74] | [0.41,0.63] | [0.36,0.58] |
| (1:4) | [0.70,0.88] | [0.67,0.87] | [0.46,0.68] | [0.43,0.65] |
| (1:3) | [0.78,0.94] | [0.77,0.93] | [0.52,0.74] | [0.49,0.71] |
| (1:2) | [0.83,0.97] | [0.82,0.96] | [0.59,0.79] | [0.55,0.77] |
| (1:1) | [0.87,0.99] | [0.90,1.00] | [0.70,0.88] | [0.69,0.87] |

Then, we proceeded by selecting the “best combined variables” among the molecular and atomic descriptors using stepwise variable selection [18]. This technique combines the advantages of the forward and backward selection procedures: at each step, a single explanatory variable may be added (forward selection) or deleted (backward elimination) from the data set. The program “*Stepwise*” of MATLAB was carried out separately for the sets of molecular and atomic descriptors. The variables providing the p -values lower than or equal to 0.001 were retained. As a result, 75 molecular and 64 atomic descriptors were selected. Thus, the combined data set of explanatory variables used in the following simulation study consisted of 139 descriptors. It is worth noting that we also carried out the polynomial SVM method with all available 1034 (209 molecular + 825 atom-pair) descriptors. However, the results obtained for such a complete set of descriptors for the (1:1) hit versus no hit and (85%/15%) training set versus test set ratios were much worse than those reported in Figure 3 (found for the combined set of 139 descriptors). For instance, the following results were obtained for the complete 1034-descriptor data set: $FN = 58$, what is 77.33% (58 of 75), and $FP = 15$, what is 20% (15 of 75), leading to the sum of errors of 97.33%. Such a poor result is certainly due to the presence in the complete data set of a large number of “noisy” variables that cannot positively contribute to machine learning process.

6 Prediction of experimental HTS data

The combined data set of 139 descriptors retained by stepwise selection [18] will be considered in this section as a basis for our simulation study intended to compare the performances of the CART, NN, SVM, LDA, PLS and kNN methods in the context of HTS. At each step, each sample was divided into the training and test subsets to build the model for prediction (for more details on the sample content, see Tables 1, 2 and 3 and Section 3). All the results presented below are the averages obtained after 100 repetitions. For the presentation of the six machine learning methods and their options being employed, the reader is referred to Section 4. The presented SVM method results were those obtained with the polynomial training function because the sums of errors it provided were usually higher compared to the linear and radial basis (rbf) functions. As in the simulations discussed in Section 5, the SVM polynomial order was set to 4.

6.1 Comparison of the CART, NN, SVM, LDA, PLS and kNN methods

Figures 3 to 5 illustrate the results generated by the six methods in terms of false negative (FN) and false positive (FP) rates, sum of errors ($FN + FP$) and method's sensitivity for the three different sample ratios presented in Tables 1, 2 and 3. For all three considered sample ratios (training/test sets: 85%/15%, 70%/30% and 50%/50%, see Section 3 for more details), polynomial SVM clearly outperformed the five other competing methods (see Figures 3 to 5).

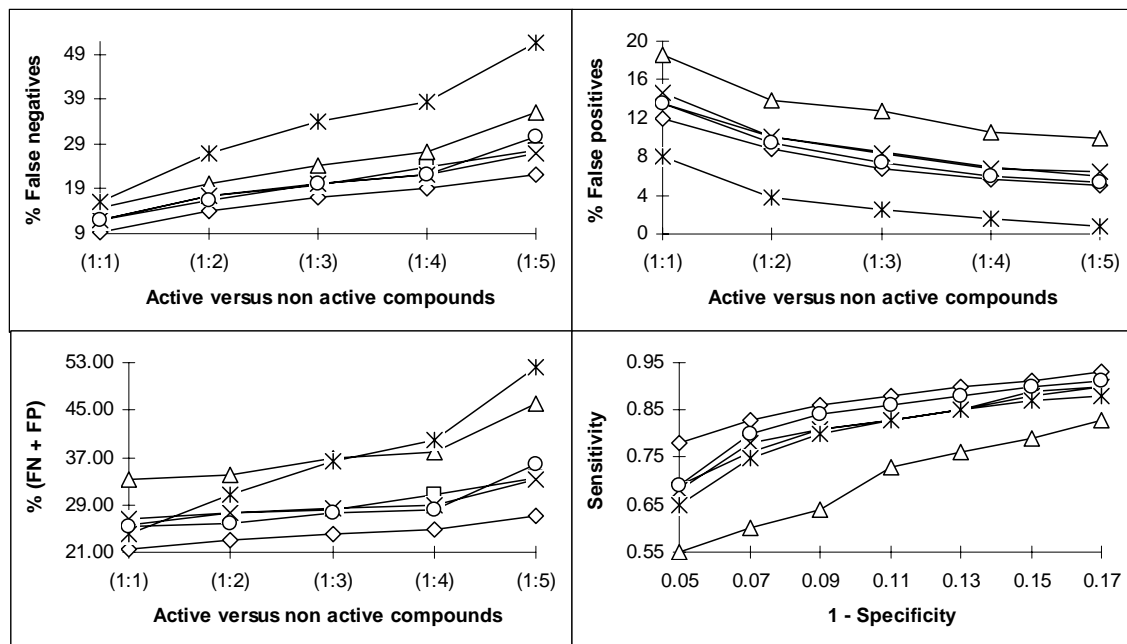


Fig. (3). Results obtained by the SVM, NN, CART, LDA, PLS and kNN methods in terms of false negatives, false positives, sum of errors and model sensitivity (ROC curve) for the proportions of the hits and no hits and the sizes of the training and test samples reported in Table 1. The SVM method results are depicted by diamonds, NN by squares, CART by triangles, LDA by crosses, PLS by asterisks and kNN by circles.

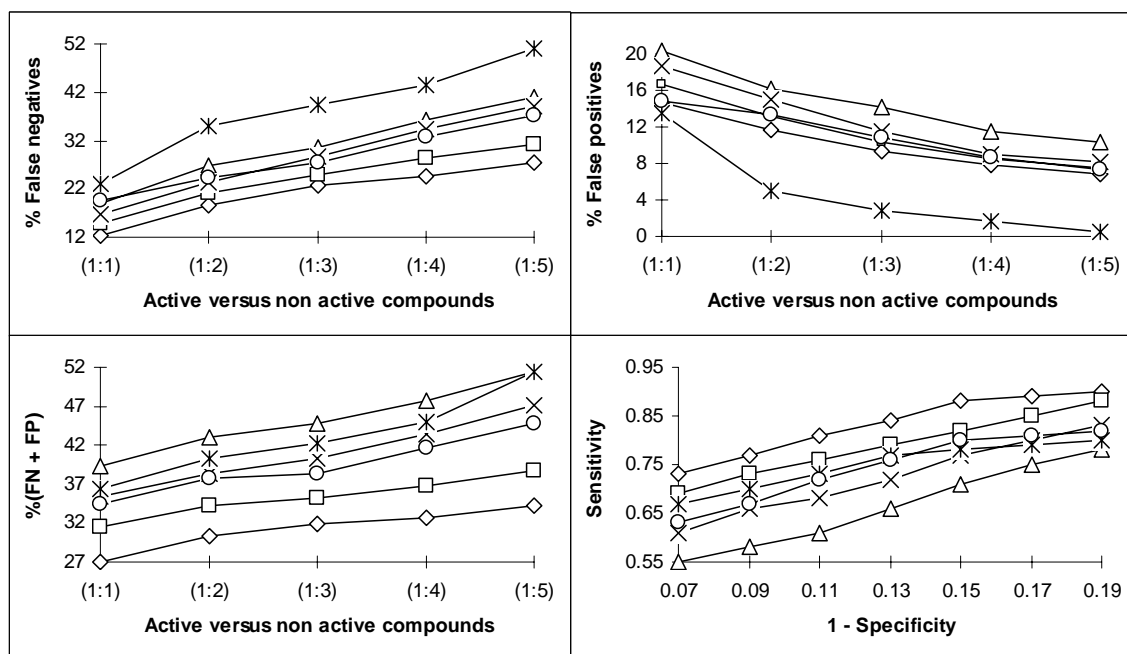


Fig. (4). Results obtained by the SVM, NN, CART, LDA, PLS and kNN methods in terms of false negatives, false positives, sum of errors and model sensitivity (ROC curve) for the proportions of the hits and no hits and the sizes of the training and test samples reported in Table 2. The SVM method results are depicted by diamonds, NN by squares, CART by triangles, LDA by crosses, PLS by asterisks and kNN by circles.

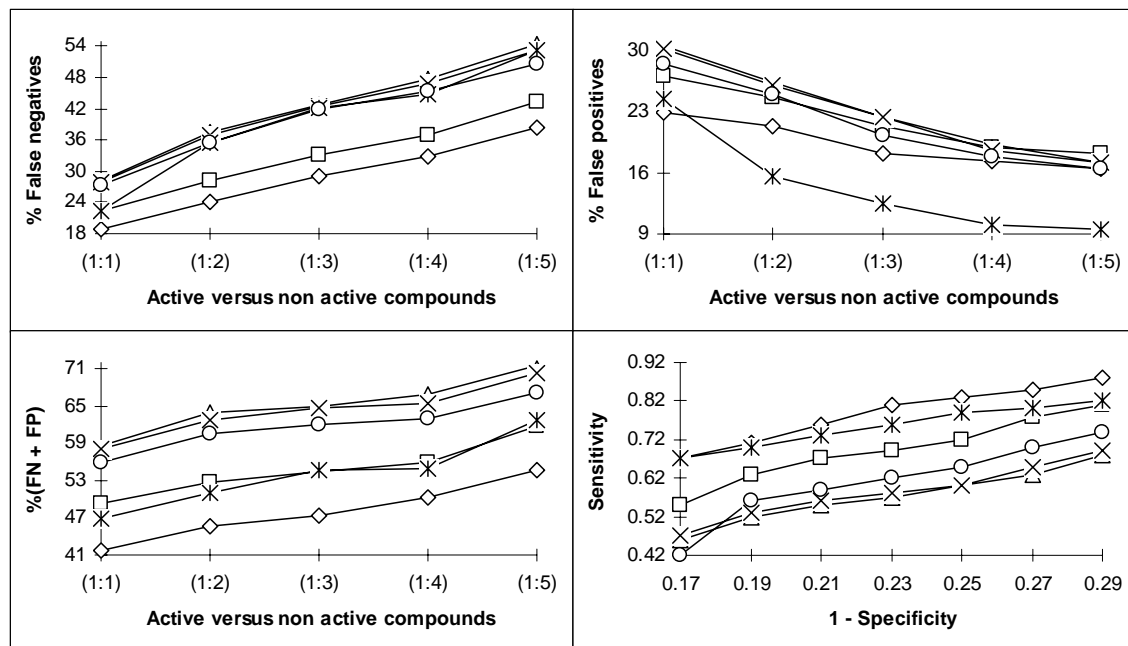


Fig. (5). Results obtained by the SVM, NN, CART, LDA, PLS and kNN methods in terms of false negatives, false positives, sum of errors and model sensitivity (ROC curve) for the proportions of the hits and no hits and the sizes of the training and test samples reported in Table 3. The SVM method results are depicted by diamonds, NN by squares, CART by triangles, LDA by crosses, PLS by asterisks and kNN by circles.

This tendency is observable for the three following parameters: the false negative rate, the sum of errors and the method's sensitivity. In terms of false negatives, the results yielded by the PLS method were the worse among the six competing methods. However, PLS produced the best overall results in terms of false positives. The PLS performance gradually deteriorates, while going from the (1:1) to (1:5) hit/no hits ratio, when we combine the false positive and false negative rates to obtain the sum of errors. The results provided by CART in terms of sensitivity were among the worse, especially for the hit/no hit ratios reported in Table 3 (Fig. 5). It is worth noting that the NN, PLS, LDA and kNN methods yielded very close results in terms of sum of errors and sensitivity performance, especially with the hit/no hit ratios reported in Tables 1 and 2 (Fig. 3 and 4). The following trend is common to all six methods: the recovery of active compounds deteriorates as their ratio in the data set decreases. Also, if the training set used in machine learning is much larger than the test set (e.g., consider 85%/15% ratio and the corresponding Fig. 3), the SVM method becomes very efficient and thus capable of making accurate hit/no hit prediction.

To confirm the results presented in Figure 3d, where the best overall results were presented, we also estimated Area Under the ROC Curves (i.e., AUC). The AUC is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one [28]. We indicated by (x_k, y_k) the coordinates of the ROC curve points ($k = 1, \dots, n$); here x_k is the false positive rate, i.e., $x_k = 1 - Sp$,

and y_k is the true positive rate, i.e., $y_k = Se$. It is worth noting that the AUC is closely related to the Gini coefficient, denoted as G and defined by the following formula [29]:

$$G + 1 = 2AUC, \quad (3)$$

where:

$$G = 1 - \sum_{k=1}^n (x_k - x_{k-1})(y_k + y_{k-1}). \quad (4)$$

The AUC is defined as follows (using Equations 3 and 4):

$$AUC = \frac{1}{2}(G + 1) = 1 - \frac{1}{2} \sum_{k=1}^n (x_k - x_{k-1})(y_k + y_{k-1}). \quad (5)$$

Table 6 reports the AUC values for the 6 machine learning methods compared in this study. As expected, the highest AUC value was obtained for the polynomial SVM method.

Table 6. The AUC values for the ROC curves representing the SVM, NN, CART, LDA, PLS and kNN methods and depicted in Figure 3d.

| <i>Method</i> | <i>SVM</i> | <i>NN</i> | <i>CART</i> | <i>LDA</i> | <i>PLS</i> | <i>kNN</i> |
|---------------|------------|-----------|-------------|------------|------------|------------|
| <i>AUC</i> | 0.946 | 0.928 | 0.894 | 0.931 | 0.925 | 0.937 |

6.2 A refined variable selection procedure in the framework of polynomial SVM

In this section we show how the variable selection can be carried out in the framework of the polynomial SVM method after the data set was portioned into the train/test splits. Such a selection is based on the sensitivity of predictors. In this experiment, the sensitivity was calculated using polynomial SVM for the 85%/15% training versus test set ratio (see Table 1) and the hit/no hit ratios varying from (1:1) to (1:5). The rationale of this approach is as follows: a single variable (i.e., descriptor) should be added to, or respectively deleted from, the data set of explanatory variables if the method's sensitivity increases after its addition, or respectively decreases after its deletion. The combined data set of 139 descriptors determined via stepwise selection [18] was used as the initial data set of explanatory variables. Then, in turn, the 134 remaining molecular descriptors ($209 - 75 = 134$) and 145 atomic descriptors ($209 - 64 = 145$) were tested for the addition to a new "optimal set of predictors". This operation was followed by a procedure intended to delete "noisy variables" from the set of optimal predictors in order to improve the sensitivity of the polynomial SVM method. After one run of this addition/deletion operation the optimal set of predictors was reduced to 88 variables. Each decision on the addition or deletion of a predictor was based on 100 prediction outcomes obtained for this predictor (i.e., 100 different training/test samples were processed to make a decision for each predictor).

Figure 6 illustrates the results obtained with such a variable selection approach and Table 7 reports the 95%-confidence intervals for the sensitivity, computed according to Equation 2, for the optimal set of 88 descriptors. For the polynomial SVM method

applied with the initial combined data set of 139 predictors (see Fig. 3 and the results presented in Section 6.1), the true positive rate ranged in the interval [0.78, 0.93], while the false positive rate varied from 5 to 17%. On the other hand, for the new optimal set of 88 descriptors retained by the discussed sensitivity-based variable selection approach, the true positive rate was in the interval [0.82, 0.94], while the false positive rate was in the 5-11% range (see Fig. 6). An average gain provided by such a variable selection procedure performed in the framework of the polynomial SVM method was about 5%.

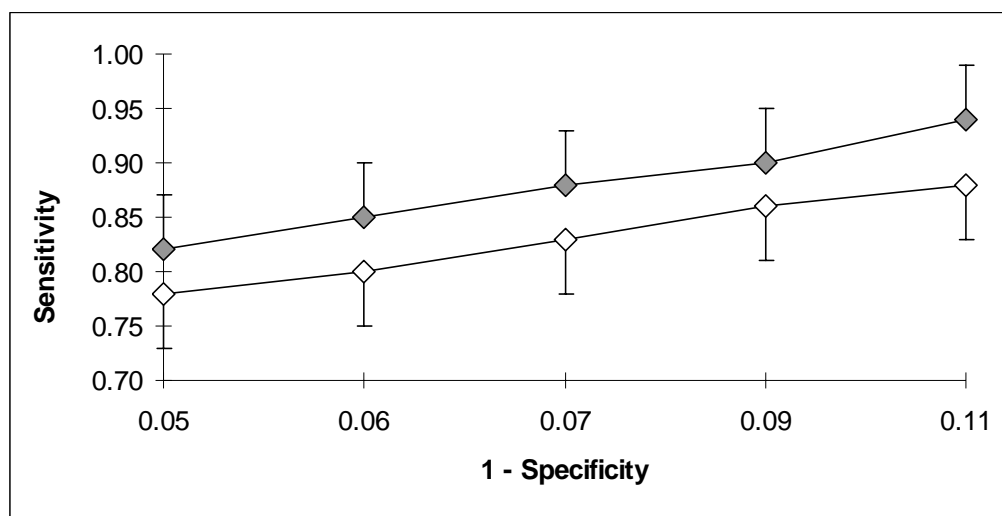


Fig. (6). ROC curves representing the results of the polynomial SVM method for the initial set of 139 descriptors (depicted by open diamonds) selected using the stepwise variable selection based on the linear regression and the optimal set of 88 descriptors (depicted by grey diamonds) selected with respect to the sensitivity of polynomial SVM. The proportions of hits and no hits and the sizes of the training and test samples were those reported in Table 1. The 139-descriptor curve is the portion of the polynomial SVM ROC curve shown in Figure 3 and corresponding to the [0.05, 0.11] interval on the x-axis. Positive error bars are shown for the curve associated with the optimal set of 88 descriptors and negative error bars for the curve associated with the initial set of 139 descriptors. The lengths of negative error bars for the former curve and positive error bars for the latter curve were very similar to the corresponding opposite direction bars shown in this figure.

Table 7. 95%-confidence intervals for the sensitivity (computed according to Equation 2) obtained with the polynomial SVM method for the optimal set of 88 predictors selected with respect to the method's sensitivity.

| Hit/No hit ratio | (1:5) | (1:4) | (1:3) | (1:2) | (1:1) |
|------------------|--------------|--------------|--------------|--------------|--------------|
| Polynomial SVM | [0.73, 0.90] | [0.77, 0.93] | [0.80, 0.95] | [0.83, 0.96] | [0.83, 0.96] |

6.3 Enrichment factor

Conventionally, so-called “enrichment factor” is used to establish a baseline for assessing the quality of virtual screening methods [13, 30]. The enrichment factor depicts the number of active compounds found by employing a virtual screening strategy, as opposed to the number of active compounds found using a random selection [30]. The machine learning methods tested in this study were evaluated by means of ROC curves. The initial

slope of a ROC curve relates to the enrichment [13]. Enrichment is calculated using the following equation:

$$\text{Enrichment} = \frac{\frac{TP}{(TP+FP)}}{\frac{P}{(P+N)}}, \quad (6)$$

where P (respectively, N) is the total number of the active (respectively, inactive) compounds in the sample, TP is the number of true positives and FP is the number of false positives. This value represents the factor by which the fraction of active compounds is increased in an *in silico* screened dataset. Table 8 summarizes the enrichment results for the test samples obtained by the six competing machine learning methods depending on the hit/no hit ratio. The training versus test set ratio used was 85% versus 15%.

Table 8. Enrichment of the test samples obtained with the polynomial SVM, NN, CART, LDA, PLS and kNN methods depending on the hit/no hit ratio.

| <i>Hit/No hit ratio</i> | <i>Pol. SVM</i> | <i>NN</i> | <i>LDA</i> | <i>CART</i> | <i>PLS</i> | <i>kNN</i> |
|-------------------------|-----------------|-----------|------------|-------------|------------|------------|
| (1:1) | 1.69 | 1.65 | 1.62 | 1.53 | 1.76 | 1.65 |
| (1:2) | 2.62 | 2.55 | 2.55 | 2.39 | 2.78 | 2.58 |
| (1:3) | 3.58 | 3.47 | 3.47 | 3.20 | 3.77 | 3.52 |
| (1:4) | 4.54 | 4.41 | 4.43 | 4.09 | 4.80 | 4.49 |
| (1:5) | 5.48 | 5.33 | 5.29 | 4.78 | 5.80 | 5.35 |

The values of the enrichment factor varied between 1.53 for CART with the ratio (1:1) and 5.80 for PLS with the ratio (1:5). The lowest enrichment was observed for the CART method for all considered hit/no hit ratios. The PLS method always provided the best enrichment. This is due to the fact that the proportions of false positives for PLS were very low, but those of false negatives very high (i.e., and those of true positives also very low). The following general trend could be noticed: the smaller is the proportion of hits in the data set, the higher is the enrichment factor.

6.4 Identification of average hits

We also examined the methods' performances with respect to the identification of the average hits from McMaster *Test data set* [1]. For this data set only 96 average hits were identified (i.e., 96 average hits, according to the terminology of the organizers of McMaster *Data Mining and Docking Competition*). Tables 9 to 11 present the identification rates of the average hits for the three considered training/test ratios (i.e., 85% versus 15%, 70% versus 30% and 50% versus 50%) provided by the polynomial SVM, NN, CART, LDA, PLS and kNN methods. The test set of each sample comprised, respectively: 15 average hits (for the results reported in Table 9), 29 average hits (for the results reported in Table 10) and 48 average hits (for the results reported in Table 11). All the results in Tables 9 to 11 are the averages calculated over 100 repetitions.

Here again, the polynomial SVM method provided the best general performance offering the most accurate identification rate of the average hits. For instance, the identification rate for the (1:1) hit/no hit ratio reported in Table 9 exceeded 75% for polynomial SVM. One can notice that the machine learning methods having high false negative rates (such as PLS and CART, see Figures 3 to 5) are also providing the lowest identification rates of the average hits. On the other hand, the SVM and LDA methods yielded very close results in terms of identifying the average hits. The methods' performances gradually decrease as the proportion of inactive compounds and the size of the test set increase. Thus, the results reported in Table 11, corresponding to the case when the test sets represented 50% of the sample and contained 48 average hits, suggest that the identification rates of the average hits were much lower in this case, especially those obtained by the PLS method.

Table 9. Percentage of average hits identified by the six considered machine learning methods depending on the hit/no hit ratio. Here, the test set represented 15% of the sample size and contained 15 of the 96 average hits.

| <i>Hit/No hit ratio</i> | <i>SVM</i> | <i>NN</i> | <i>CART</i> | <i>LDA</i> | <i>PLS</i> | <i>kNN</i> |
|-------------------------|------------|-----------|-------------|------------|------------|------------|
| (1:1) | 75.40 | 72.13 | 46.47 | 74.33 | 27.47 | 71.27 |
| (1:2) | 60.20 | 54.60 | 32.13 | 60.40 | 15.80 | 65.13 |
| (1:3) | 56.33 | 52.73 | 26.87 | 54.80 | 8.20 | 57.87 |
| (1:4) | 47.07 | 47.20 | 25.93 | 41.20 | 5.07 | 50.60 |
| (1:5) | 35.73 | 31.47 | 23.07 | 35.67 | 3.87 | 40.47 |

Table 10. Percentage of average hits identified by the six considered machine learning methods depending on the hit/no hit ratio. Here, the test set represented 30% of the sample size and contained 30 of the 96 average hits.

| <i>Hit/No hit ratio</i> | <i>SVM</i> | <i>NN</i> | <i>CART</i> | <i>LDA</i> | <i>PLS</i> | <i>kNN</i> |
|-------------------------|------------|-----------|-------------|------------|------------|------------|
| (1:1) | 59.40 | 50.60 | 29.20 | 55.20 | 19.60 | 57.50 |
| (1:2) | 36.73 | 30.17 | 22.17 | 38.87 | 8.47 | 39.07 |
| (1:3) | 30.87 | 29.27 | 19.63 | 34.83 | 4.37 | 33.60 |
| (1:4) | 19.80 | 17.33 | 13.57 | 30.17 | 1.40 | 27.90 |
| (1:5) | 14.83 | 10.20 | 7.23 | 20.07 | 0.30 | 24.27 |

Table 11. Percentage of average hits identified by the six considered machine learning methods depending on the hit/no hit ratio. Here, the test set represented 50% of the sample size and contained 48 of the 96 average hits.

| <i>Hit/No hit ratio</i> | <i>SVM</i> | <i>NN</i> | <i>CART</i> | <i>LDA</i> | <i>PLS</i> | <i>kNN</i> |
|-------------------------|------------|-----------|-------------|------------|------------|------------|
| (1:1) | 48.63 | 42.40 | 25.54 | 48.02 | 14.19 | 47.60 |
| (1:2) | 37.33 | 32.31 | 20.10 | 35.75 | 7.52 | 32.04 |
| (1:3) | 27.19 | 24.38 | 16.42 | 26.21 | 2.96 | 22.94 |
| (1:4) | 17.60 | 13.54 | 11.21 | 16.31 | 1.42 | 18.15 |
| (1:5) | 10.65 | 8.79 | 5.60 | 9.63 | 0.17 | 11.33 |

7 Conclusion and future work

As a traditional high-throughput screening campaign remains a very costly process, the development of *in silico* methods allowing one to predict accurately experimental HTS

results would be of great importance. The current study suggests that the machine learning methods can be successfully applied in HTS and shows the ways for improving their performance. It also highlights the main limitations of machine learning techniques in the context of HTS. In general, the results provided by the six considered machine learning methods, including SVM, NN, CART, LDA, PLS and kNN, in terms of sensitivity were very encouraging, especially those obtained by polynomial SVM, for the case when the training set represented 85% and the test set 15% of the sample size.

First, we compared between them the sets of atomic and molecular descriptors characterizing each considered chemical compound (Fig. 2) in order to determine which of these types of descriptors provides a better discrimination of the hit/no hit outcomes. To the best of our knowledge, such a comparison, made in the context of HTS, is novel. We showed that the atomic and molecular descriptors had generally the same power in predicting the active molecules when the NN and LDA methods were applied. When the linear and especially polynomial SVM methods were performed, the molecular descriptors provided a better prediction than their atomic counterparts. However, the confidence intervals presented in Tables 4 and 5 suggest that the obtained prediction differences were not significant.

Second, we carried out a stepwise regression to select the best descriptors in both data sets and create a combined set of explanatory variables. The six considered machine learning methods were then tested on this combined set of variables and their performances were evaluated in terms of false negatives, false positives, sum of errors and sensitivity. The methods comparison was carried out for three different training/test set ratios and five different hit/no hit proportions. The conducted simulations suggest that the polynomial SVM method outperforms NN, CART, LDA, PLS and kNN in most circumstances. The reported results also demonstrate that the CART and PLS methods are not very efficient for predicting the active molecules, especially when the ratio of the active compounds is very marginal compared to that of the inactive ones. The bad performances of the CART method can be due to the fact that the CART predictions are based on the mean activity of the training compounds in the final leaves of the decision tree. It can be also due to the enormous reduction of feature space carried out by CART. As to PLS, we noticed that the proportions of false positives provided by this method were very low, but the proportions of false negatives were very high compared to the other competing methods.

We also implemented a variable selection procedure based on the sensitivity of a machine learning method. This procedure consists of the deletion of a variable, if the method's sensitivity decreases, or of its addition, if the method's sensitivity increases. Figure 6 shows the performances of this approach obtained in the framework of the polynomial SVM method. An average gain provided by such a refined variable selection procedure was about 5% (Fig. 6).

In the future, we plan, firstly, to apply classification methods allowing one to eliminate the descriptors not contributing to clustering (i.e., noisy variables), secondly, to use information from computational and combinatorial chemistry in order to improve the prediction accuracy (i.e., add to the set of descriptors the docking and binding scores which can be computed by many cheminformatics software; e.g., see the software comparison article by Moitessier et al. [31]), and, finally, to deploy a 2-fold machine learning procedure for large HTS data sets having small percentages of hits (such a

procedure could consider one set of descriptors to preselect 10-15% of the best compounds at the first step, and then perform a second selection of the active compounds, using a different set of descriptors, within the preselected set of compounds). Another area for future investigation consists of a joint application of the most powerful machine learning approaches. For instance, a combined application of the polynomial SVM and NN methods could be explored.

Acknowledgements

The authors thank FQRNT (Quebec Research Funds in Nature and Technology) for funding this project. The authors are grateful to the Ph.D. student Dunarel Badescu for his help in the analysis of chemical data.

References

- [1] Elowe, N. H.; Blanchard, J. E.; Cechetto, J. D.; Brown, E. D. Experimental screening of DHFR yields a test set of 50,000 small molecules for a computational data-mining and docking competition. *J Biomol Screen* **2005**, *10*, 653-657.
- [2] Brideau, C.; Gunter, B.; Pikounis, W.; Liaw, A. Improved statistical methods for hit selection in high throughput screening. *J Biomol Screen* **2003**, *8*, 634-647.
- [3] Kevorkov, D.; Makarenkov, V. Statistical analysis of systematic errors in high throughput screening. *J Biomol. Screen* **2005**, *10*, 557-567.
- [4] Malo, N.; Hanley, J. A.; Cerquozzi, S.; Pelletier, J.; Nadon, R. Statistical practice in high throughput screening data analysis. *Nat Biotech* **2006**, *24*, 167-175.
- [5] Makarenkov, V.; Kevorkov, D.; Gagarin, A.; Zentilli, P.; Malo, N.; Nadon, R. An efficient method for the detection and elimination of systematic error in high throughput screening. *Bioinformatics* **2007**, *23*, 1648-1657.
- [6] Briem, H.; Günther, J. Classifying “Kinase Inhibitor-Likeness” by using machine-learning methods. *Chem Bio Chem* **2005**, *6*, 558-566.
- [7] Müller, K. R.; Rätsch, G.; Sonnenburg, S.; Mika, S.; Grimm, M.; Heinrich, N. Classifying “drug-likeness” with kernel-based learning methods. *J Chem Inf Model* **2005**, *45*, 249-253.
- [8] Burton, J.; Ijjaali, I.; Barberan, O.; Petitet, F.; Vercauteren, D. P.; Michel, A. Recursive partitioning for the prediction of cytochromes P450 2D6 and 1A2 inhibition: importance of the quality of the dataset. *J Med Chem* **2006**, *49*, 6231-6240.
- [9] Harper, G.; Pickett, S. Methods for mining HTS data. *Drug Disc Today* **2006**, *11*, 694-699.
- [10] Plewczynski, D.; Von, G. M.; Spieser, S. A.; Rychlewski, L.; Wyrwicz, L. S.; Ginalska, K.; Koch, U. Target specific compounds identification using a support vector machine. *Comb Chem and HTS* **2007**, *10*, 189-196.
- [11] Simmons, K.; Kinney, J.; Owens, A.; Kleier, D.; Bloch, K.; Argentar, D.; Walsh, A.; Vaidyanathan, G. Comparative study of machine learning and chemometric tools for analysis of in-vivo high throughput screening data. *J Chem Inf Model* **2008a**, *48*, 1663-1668.
- [12] Simmons, K.; Kinney, J.; Owens, A.; Kleier, D.; Bloch, K.; Argentar, D.; Walsh, A.; Vaidyanathan, G. Practical Outcomes of Applying Ensemble Machine Learning Classifiers to High-Throughput Screening (HTS) Data Analysis and Screening. *J*

- Chem Inf Model* **2008b**, 48, 2196-2206.
- [13] Butkiewicz, M.; Mueller, R.; Selic, D.; Dawson, E.; Meiler, J. Application of machine learning approaches on Quantitative Structure Activity Relationships. *IEEE* **2009**, pp 255-262.
 - [14] Fang, J.; Dong, Y.; Lushington, G. H.; Ye, Q. Z.; Georg, G. I. Support Vector Machines in HTS data mining: Type I MetAPs inhibition study. *J Biomol Screen* **2006**, 11, 138-144.
 - [15] Todeschini, R.; Consonni, V. *Handbook of Molecular Descriptors*. Wiley-VCH, **2000**.
 - [16] Nebel, J. C. Generation of 3D templates of active sites of proteins with rigid prosthetic groups. *Bioinformatics* **2006**, 22, 1183-1189.
 - [17] Molecular Operating Environment (MOE), version 2008 *Chemical Computing Group*, Montreal, Quebec, Canada, **2008**.
 - [18] Hogarty, K.; Kromrey, J.; Ferron, J.; Hines, C. Selection of variable in exploratory factor analysis: An empirical comparison of a stepwise and traditional approach. *Psychometrika* **2004**, 69, 593-611.
 - [19] Breiman, L.; Friedman, J. H.; Stone, R. A.; Olshen, C. J. *Classification And Regression Trees*. Chapman and Hall, New York, **1984**.
 - [20] Haykin, S. *Neural networks: a comprehensive foundation*. Prentice Hall, **1999**.
 - [21] Rumelhart, D. E.; Hinton, G. E.; Williams, R. J. Learning internal representations by error propagation. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition* **1986**, MIT Press, Cambridge.
 - [22] Vapnik, V. *Statistical learning theory*. Willey, **1998**.
 - [23] Shakhnarovich, G.; Darrell, T.; Indyk, P. Nearest-Neighbor Methods in Learning and Vision: Theory and Practice. MIT Press, **2006**.
 - [24] McLachlan, G. J. *Discriminant Analysis and Statistical Pattern Recognition*. Wiley Interscience, **2004**.
 - [25] Helland, I. On the structure of partial least squares regression. *Com Stat Sim Comp* **1988**, 17, 581-607.
 - [26] Frank, I.; Friedman, J. A statistical view of some chemometrics regression tools. *Technometrics* **1993**, 35, 109-135.
 - [27] Fawcett, T. Using rule sets to maximize ROC performance. In *Proceedings of the IEEE International Conference on Data Mining* **2001**, pp 131-138.
 - [28] Fawcett, T. An introduction to ROC analysis. *Pattern Recognition Letters* **2006**, 27, 861-874.
 - [29] Hand, D. J., & Till, R. J. A simple generalization of the area under the ROC curve to multiple class classification problems. *Machine Learning* **2001**, 45, 171-186.
 - [30] Bender, A.; Glen, R. C. A discussion of measures of enrichment in virtual screening: comparing the information content of descriptors with increasing levels of sophistication. *J Chem Inf Model* **2005**, 45, 1369-1375.
 - [31] Moitessier, N.; Englebienne, P.; Lee, D.; Lawandi, J.; Corbeil, C. R. Towards the development of universal, fast and highly accurate docking/scoring methods: a long way to go, *Brit J Pharm* **2008**, 153, S7-S26.